

OEO041

Programmazione Java SE avanzato

Durata: 5 gg

Descrizione

Questo corso copre le caratteristiche essenziali del linguaggio Java SE 8 e delle sue API - Application Program Interfaces, che vengono utilizzate per la progettazione e costruzione delle applicazioni Object Oriented con la piattaforma Java Standard Edition 8. In questo corso si impara a creare le applicazioni object-oriented con l'ultima tecnologia JDK seguendo le migliori pratiche del linguaggio. Questo corso è propedeutico alla preparazione all'esame Oracle Certified Professional, Java SE 8 Programmer.

Le competenze acquisite includono:

- Progettare correttamente le classi Java
- Utilizzare le espressioni lambda nelle applicazioni Java seguendo il paradigma della programmazione funzionale
- Utilizzare il Java Collections Framework
- Gestire file, directory e file system.
- Connettersi ai database relazionali e utilizzare i tre tipi fondamentali di query SQL con la tecnologia JDBC.
- Scrivere le applicazioni multi-threaded ad alte prestazioni.

A chi è rivolto?

Il corso è rivolto a sviluppatori Java

Prerequisiti

Conoscenza base del linguaggio Java SE 8.

Contenuti

Progettazione avanzata delle classi

- Revisione dei concetti Oracle Certified Associate
- Modificatori d'accesso
- Overloading e Overriding
- Classi astratte
- Static e Final
- Gli Import
- Utilizzo di instanceof
- Capire invocazione dei metodi virtuali
- Annotare i metodi sovrascritti
- Sovrascrivere equals, hashCode, e toString
- Lavorare con le Enums
- Utilizzare Enums nelle istruzioni Switch
- Aggiungere costruttori, campi e metodi ad una classe
- Creare le classi innestate

- Classi-membri interne
- Classi interne locali
- Classi interne anonime
- Classi innestate static

Utilizzare i Design Patterns

- Progettare un'interfaccia, obiettivi
- Introduzione alla programmazione funzionale
- Definire un'interfaccia funzionale
- Implementare le interfacce funzionali con le espressioni lambda
- Applicare l'interfaccia Predicate
- Implementare il polimorfismo
- Differenza fra un oggetto e un riferimento
- Conversione del tipo (casting) di riferimenti a oggetti
- Capire i principi di progettazione
- Incapsulare i dati
- Creare i JavaBean
- Applicare una relazione e-un (is-a)
- Applicare la relazione ha-un (has-a)
- Composizione degli oggetti
- Lavorare con i Design Pattern
- Applicare il pattern Singleton
- Creare gli oggetti immutabili
- Utilizzare il pattern Builder
- Creare gli oggetti con il pattern Factory
- Generics e Collections
- Array e ArrayList
- Ordinamento e ricerca
- Classi Wrapper, autoboxing e unboxing
- L'operatore diamante (diamond operator)
- Lavorare con i generics: classi, interfacce e metodi generici
- Interagire con il codice legacy
- I limiti (bounds)
- Utilizzare le interfacce List, Set, Map e Queue
- Metodi comuni della classe Collections
- Confronto fra vari tipi di collezioni
- Comparator vs. Comparable
- Utilizzare i riferimenti ai metodi
- Rimozione condizionale
- Aggiornare tutti gli elementi
- Iterare sulle collezioni

Programmazione funzionale

- Utilizzare variabili nelle espressioni lambda
- Lavorare con le interfacce funzionali incorporate
- Implementare Supplier
- Implementare Consumer and BiConsumer

- Implementare Predicate and BiPredicate
- Implementare Function and BiFunction
- Implementare UnaryOperator and BinaryOperator
- Verificare le interfacce funzionali
- Restituire Optional
- Utilizzare i flussi (stream)
- Creare le sorgenti dei flussi
- Utilizzare le operazioni comuni sul terminale
- Utilizzare le operazioni comuni intermedie
- Collegare i pipeline
- Stampare un flusso
- Creare i flussi primitivi
- Lavorare con i flussi Pipeline
- Collegare i flussi ai dati sottostanti
- Concatenare gli Optional
- Raggruppare i risultati

Eccezioni e asserzioni

- Revisione delle eccezioni
- Terminologia delle eccezioni
- Categorie delle eccezioni
- Istruzione try
- Throw vs. Throws
- Creare le eccezioni personalizzate
- Catturare le eccezioni multiple
- Utilizzare try-with-resources
- AutoCloseable
- Le eccezioni sopresse
- Rilanciare le eccezioni
- Istruzione assert
- Abilitare e usare le asserzioni

Esecuzione concorrente

- Introduzione ai thread
- Distinguere fra i tipi di thread
- Capire la concorrenza dei thread
- Utilizzare la classe Thread e l'interfaccia Runnable
- Il metodo sleep()
- Creare i thread con ExecutorService
- Introduzione al Single-Thread Executor
- Chiudere un Thread Executor
- Sottoporre i task, aspettare i risultati, schedare i task
- Potenziare la concorrenza con i pool
- Sincronizzare l'accesso ai dati
- Proteggere i dati con le classi atomiche
- Migliorare l'accesso con i blocchi sincronizzati
- Sincronizzazione dei metodi

- Capire il costo della sincronizzazione
- Utilizzare le collezioni concorrenti
- Capire gli errori di consistenza della memoria
- Lavorare con le classi concorrenti
- Ottenete le collezioni sincronizzate
- Lavorare con i flussi (stream) paralleli
- Gestire i processi concorrenti
- Creare un CyclicBarrier
- Applicare il Fork/Join Framework
- Identificare i problemi di threading
- Capire liveness
- Gestire le condizioni di competizione (race conditions)

IO

- Capire file e directory
- Concetti di File System
- Classe File
- Introduzione ai flussi, tipi di flussi
- Operazioni comuni sui flussi
- Lavorare con i flussi
- Le classi FileInputStream e FileOutputStream
- Le classi FileReader e FileWriter classes
- Le classi ObjectInputStream e ObjectOutputStream
- Le classi PrintStream e PrintWriter
- Interazione con l'utente: vecchio e nuovo stile

NIO.2

- Introduzione a NIO.2
- Introduzione a Path
- Creare i Path
- Interagire con i Path e File
- Fornire argomenti opzionali
- Utilizzare gli oggetti Path
- Capire gli attributi dei file
- Migliorare l'accesso con le View
- Nuovi metodi di Stream
- Attraversare una directory
- Fare la ricerca in una directory
- Visualizzare il contenuto di una directory
- Stampare il contenuto di un file
- Confronto fra i metodi File obsoleti e NIO.2

JDBC

- Introduzione ai database relazionali e SQL
- Scrivere le istruzioni SQL di base
- Introduzione alle interfacce JDBC
- Connettersi ad un database
- Costruire database URL

- Ottenere un oggetto Statement
- Scegliere il tipo di ResultSet e il modo di accesso concorrentiale
- Eeguire uno Statement
- Recuperare i dati da un ResultSet
- Chiudere le risorse JDBC
- Gestire le eccezioni JDBC